Extension of the OpenMC depletion module for transport-independent depletion

Oleksandr Yardas[1] ○ ✉, **Paul Romano**[2] ○, **Madicken Muk**[3] ○, and **Kathryn Huff**[1] ○

[1]Dept. of Nuclear, Plasma, and Radiological Engineering, University of Illinois Urbana-Champaign, [2]Argonne National Laboratory, [3]Dept. of Nuclear Science and Engineering, Oregon State University Yet

Abstract

We have added functionality for running depletion simulations independently of neutron transport in OpenMC, an open source Monte Carlo particle transport code with an internal depletion module. Transport-independent depletion uses pre-computed static multigroup cross sections and fluxes to calculate reaction rates for OpenMC's depletion matrix solver. This accelerates the depletion calculation, but removes the spatial coupling between depletion and neutron transport.

We used a simple PWR pincell to validate the method against the existing transport-coupled depletion method. Nuclide concentration errors roughly scale with depletion time step size and are inversely proportional to the amount of the nuclide present in a depletable material. The magnitude of concentration error depends on the nuclide of interest. Concentration errors for low-abundance nuclides at longer (30-day) time steps exhibit large negative initial concentration the becomes more positive with time due to overestimation of nuclide production stemming from the lack of spatial coupling to neutron transport. For ten 3-day time steps, fission product concentration errors are all under 3%. Actinide concentration errors range from 10-15% for Am and Cm, 5-7% for Pu and Np, and 2% and less for U. Surprisingly, the numbers are similar for 30-day time steps. These results demonstrate the potential of this new method with moderate accuracy and extraordinary time savings for low and medium fidelity simulations. Concentration error characterization on larger models remains an open area of interest.

Keywords depletion, coupling, OpenMC, Cross sections

1. Introduction

Accurate simulations of advanced reactors on exascale computing hardware requires state-of-the-art software tools in Monte Carlo transport and heat-coupled computational fluid dynamics [1], [2]. These methods are computationally expensive, but deliver high confidence in their results. If any component in this tool chain could be simplified or sped up, it would reduce the computational cost on and economic cost from these exascale machines. For Monte Carlo neutron transport, depletion simulations are one of the more expensive simulations. *Depletion* is the process by which the nuclide composition changes due to nuclear transmutation and decay reactions occurring in a material subject to irradiation. Depletion capabilities were recently added [3] to OpenMC [4], an open source, community developed Monte Carlo particle transport code. The depletion solver utilizes OpenMC's Python API to iteratively run a transport simulation, calculate transmutation reaction rates, solve the depletion matrix (Eq. 2), and update the material composition. This makes transport-coupled depletion calculations computationally expensive, especially for large reactor models.

In the present work, we describe a new method for running depletion calculations without the need to iteratively run transport calculations by using pre-calculated microscopic cross sections and fluxes for each depletion step. At most, only one transport calculation needs to be run to generate these cross sections and fluxes. Neutron transport and depletion are tightly coupled because the change in material composition can result in changes to the spatial distribution of reaction rates. If few materials are used in a transport-independent depletion calculation, this effectively removes spatial effects of the changing material composition due to the static fluxes and cross sections.

This limits the practical application of transport-independent depletion to domains where capturing spatial effects are a secondary concern. One such application is in global fuel-cycle analysis where we are more concerned with the overall change in material composition. Fuel cycle simulations typically model many different reactors simultaneously over long time periods. The capabilities described in this paper have been utilized in Cyclus [5] – an open source fuel cycle simulator – for coupling depletion to the fuel cycle simulaton [6]. Another area of application is in facilities where the coupling between neutron transport and depletion is effectively one-way. This is often the case for experimental facilities with a neutron source where the source rate is high enough to cause activation of materials but not high enough to produce meaningful burnup of materials. Many proposed fusion energy facilities would fall into this category. For such applications, it is unnecessary to run a fully coupled simulation between neutron transport and depletion; instead, a neutron transport simulation corresponding to the beginning of irradiation can determine reaction rates relative to the neutron source rate, and then the depletion calculation can be performed using the reaction rates scaled by any changes in the neutron source rate over time. For fusion systems, these calculations are often carried out with FISPACT-II [7]; See [8] for a recent example. The capabilities described in this paper have also been utilized for a recent study of shutdown dose rate in fusion systems [9].

This paper is organized as follows: In Methods, we describe the methods and algorithms used to calculate reaction rates using the new depletion capabilities. In Results, we describe our simulation and analyze the results. In Conclusion, we summarize our results, and discuss some gaps in the current implementation of the new feature.

2. METHODS

Depletion is typically done in sub-regions that are small enough to have little spatial variation in the flux (and hence in the reaction rates). For example, in a model of a full reactor core, depletion regions may correspond to individual fuel pins, or even smaller sub-pin regions of radial rings and azimuthal segments. The general form of the equation governing depletion is

$$
\frac{dN_i}{dt} = \sum \mathrm{limits}_{j=1}^{n} \left[\int_0^\infty f_{j \to i}(E) \sigma_j(E) \phi(E,t) dE + \lambda_{j \to i} \right] N_j(t)
$$
$$
- \left[\int_0^\infty \sigma_i(E) \phi(E,t) dE + \sum \mathrm{limits}_{j=1}^{n} \lambda_{i \to j} \right] N_i(t),
$$

(1)

where

$$N_i(t) \equiv \text{density of nuclide i at time t}$$
$$\sigma_i(E) \equiv \text{transmutation cross section for nuclide i at energy E}$$
$$\phi(E, t) \equiv \text{neutron flux at energy E and time t}$$
$$f_{j \to i}(E) \equiv \text{fraction of transmutation reactions in nuclide j that produce nuclide i} \tag{2}$$
$$\lambda_{j \to i} \equiv \text{decay constant for decay modes in nuclide j that produce nuclide i}$$
$$n \equiv \text{total number of nuclides.}$$

The summations in Equation (1) are effectively over all nuclides except i, as $f_{i \to i}(E)$ and $\lambda_{i \to i}$ must be zero by definition. Equation (1) can be condensed into a matrix-vector equation:

$$\frac{d\boldsymbol{n}}{dt} = \boldsymbol{A}(\boldsymbol{n}, t)\boldsymbol{n} \tag{3}$$

where

$$\boldsymbol{n} = \begin{pmatrix} N_1 \\ N_2 \\ \vdots \\ N_n \end{pmatrix}, \tag{4}$$

and $\boldsymbol{A}(\boldsymbol{n}, t) \in \mathbb{R}^{n \times n}$ is the depletion matrix containing the decay and transmutation coefficients. The coefficients of $\boldsymbol{A}$ change with time and depend on the nuclide composition vector $\boldsymbol{n}$, making this a nonlinear system of equations. OpenMC utilized the CRAM method to solve this system of equations [3].

We had two design goals for transport-independent depletion functionality:

1. Enable use of OpenMC depletion capabilities using only the Python API
2. Enable depletion of materials directly given a multi-group flux for each material.

We created the `IndependentOperator` and `MicroXS` classes to achieve these design goals.

2.1. MicroXS

The `MicroXS` class contains cross section data indexed by nuclide, reaction name, and energy group. It also contains methods to import and export `MicroXS` objects from `.csv` files. In principle, a user could create multi-group microscopic cross sections with a transport solver code, and then use the `MicroXS` class to read in the cross section data for use in depletion. However, this is unnecessary as the `get_microxs_and_flux()` function runs an OpenMC k-eigenvalue calculation to create a `MicroXS` object and multi-group flux profile for each user-specified domain.

In the initial release of this feature in OpenMC v0.13.1, `MicroXS` sub-classed the `pandas.DataFrame.` class to store data and assumed a one-group structure. The v0.14.0 release removed the `pandas` dependency and refactored `MicroXS` class to store multi-group cross section data.

2.2. IndependentOperator

The `Operator` class contains data and methods for running transport-coupled depletion calculations and runtime processing of OpenMC statepoint files to extract the data needed to solve Equation (2). In this work, we refactored the `Operator` class into a `CoupledOperator` class maintaining the existing depletion capability. We implemented the transport-independent depletion machinery in a new class, `IndependentOperator` which uses an instance of the `MicroXS` class and corresponding fluxes for each depletion domain to calculate reaction rates for that domain using multi-group microscopic cross sections.

Calculating reactions rates using multi-group cross sections is mathematically equivalent to tallying the reaction rates directly. In transport-coupled depletion calculation, the transmutation reaction rates $r_{ij}(t)$ are tallied directly as:

$$r_{ij}(t) = \left[\int_0^\infty \sigma_{ij}(E)\phi(E,t)dE \right] N_i(t) \tag{5}$$

where i and j indicate the nuclide and reaction, respectively.

To perform depletion using multi-group cross sections, the multi-group cross sections and fluxes are multiplied to get a per-source neutron reaction rate:

$$r_{ij}(t) = \left[\sum_g \sigma_{ij,g}\phi_g(t) \right] N_i(t) \tag{6}$$

In practice, this introduces a discretization error, but in the limit of infinitely small energy groups, Equation (5) is equivalent to Equation (4). To get a time-dependent microscopic cross section or flux, we must perform transport calculations (and in the case of time-varying microscopic cross sections, incorporate thermal feedbacks). Herein lies the primary assumption of transport-independent depletion: *microscopic cross sections and reactor fluxes are static.*

Both the tallied and multi-group-calculated reaction rates have units of reactions/ src particle. A normalization factor f in src/s is applied to obtain the more conventional unit of reactions/s:

$$R_{ij} = r_{ij}f \tag{7}$$

This normalization factor is typically related to the power of the reactor and energy released from fission (`fission-q` normalization) or the external source strength (`source-rate` normalization). `IndependentOperator` may use either of these methods to normalize the reaction rates. While `CoupledOperator` may use several methods to calculate fission yields, including using the Monte Carlo simulation to directly tally fission yields, `IndependentOperator` relies on constant yield data included in the depletion chain file.

3. RESULTS

To validate transport-independent depletion, we ran depletion simulations on a PWR with a single depletion zone for three cases:

1. Transport-coupled depletion,
2. Transport-independent depletion,
3. Transport-independent depletion with microscopic cross sections updated after each depletion step.

The first case serves as a reference solution we use to estimate the error in the solution of the new method. We used 25 inactive and 125 active batches, with 10^6 particles per batch. For the second case, we used three different group structures to test the effect on solution accuracy: one group, CASMO-8, and CASMO-40. The third case is much slower than the first case as the cross sections need to be reloaded after each depletion step. The third case is a sanity check that the cross sections are being computed correctly, as it should have a smaller error than the second case. Due to the cross section discretization that occurs before performing depletion, we do not expect the third case results to converge to the first case results in limit of infinite particles, but we would expect it to converge in the limit of infinite particles *and* energy groups. The third case was only run for one-group transport-independent depletion.

Table 1 and Table 2 contain the material parameters and compositions of our model, and Table 3 contains the geometric parameters. We used the ENDF/B-VII.1 nuclear data library available at openmc.org/data. We used the ENDF/B-VII.1 depletion chain in the PWR spectrum also available at openmc.org/data.

Table 1. *Material Parameters*

Item	Fuel	Cladding	Water
Density [g cm^{-3}]	10.4	6	1.0
Volume [cm^3]	0.1764π	–	–
S(α,β)	–	–	'c_H_in_H2O'

Table 2. *Material Compositions*

Fuel composition		Cladding composition		Water composition	
Nuclide	Composition [atom %]	Nuclide	Composition [atom %]	Nuclide	Composition [atom %]
^{15}O	0.000758	^{90}Zr	0.5145	^{1}H	1.999689
^{16}O	1.999242	^{91}Zr	0.1122	^{2}H	0.000311
^{234}U	0.000385	^{92}Zr	0.1715	^{15}O	0.999621
^{235}U	0.043020	^{94}Zr	0.1738	^{16}O	0.000379
^{236}U	0.000197	^{96}Zr	0.028		
^{238}U	0.956398				

Table 3. *Geometric Parameters*

Fuel Radius [cm]	Clad Radius [cm]	Water Bounding Box dimensions [cm $\times$ cm]
0.42	0.45	1.24×1.24

Depletion is a slow process whose effects on neutronics only start to apply on longer timescales (days and months). For validation purposes we ran each of the three cases with four different magnitudes of timestep size: 360 seconds, 4 hours, 3 days, and 30 days. All simulations ran for 10 depletion steps. We used the `PredictorIntegrator` time stepper, which is analogous to the Euler method. Predictor-corrector methods are not useful for transport independent depletion. The correction step will utilize the same reaction rates as the initial prediction due to the static fluxes and cross sections. This means any predictor-corrector method will produce the same numerical results as a predictor method but with higher floating-point errors due to the increased number of operations[1]. We used constant reaction rate Q values multiplied by the reaction rates to obtain the normalization factor f (`fission-q` normalization) for all cases with a linear power density of 174 W/cm.

The general trend in our results is that concentration errors are smaller for shorter depletion time steps, and larger for longer depletion time steps, however some nuclides exhibit more complicated behavior.

---

[1]Preliminary testing with predictor-corrector methods confirmed the higher floating point errors, however their magnitudes are marginal.
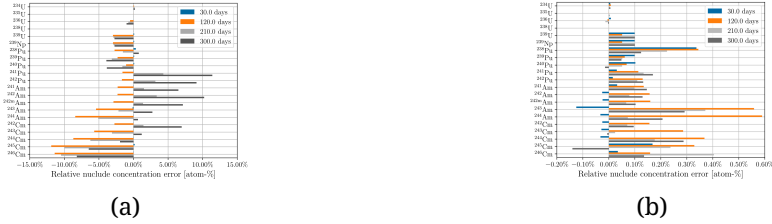
Figure 1. *Relative actinide concentration error using 3-day time steps at 3, 12, 21, and 30 days of depletion for (a) constant cross sections; (b) updating cross sections.*

Figure 1a and Figure 1b show the relative error in predicted actinide concentration for constant cross sections and updating cross sections, respectively, using 3-day time steps. Figure 3a and Figure 3b show the same respective quantities for 30-day time steps. As expected, updating the cross sections at each depletion step results in low predicted nuclide concentration errors, on the order of a fraction of a percent. The error trend for constant cross sections depends both on the cross section and time step size. For example, constant cross sections using a 3-day time step underpredicts the ^{241}Pu concentration, but at longer time steps overpredicts the concentration. Figure 2 shows the overprediction is due to overcalculating the rate of (n, γ) reactions that occur on ^{240}Pu.
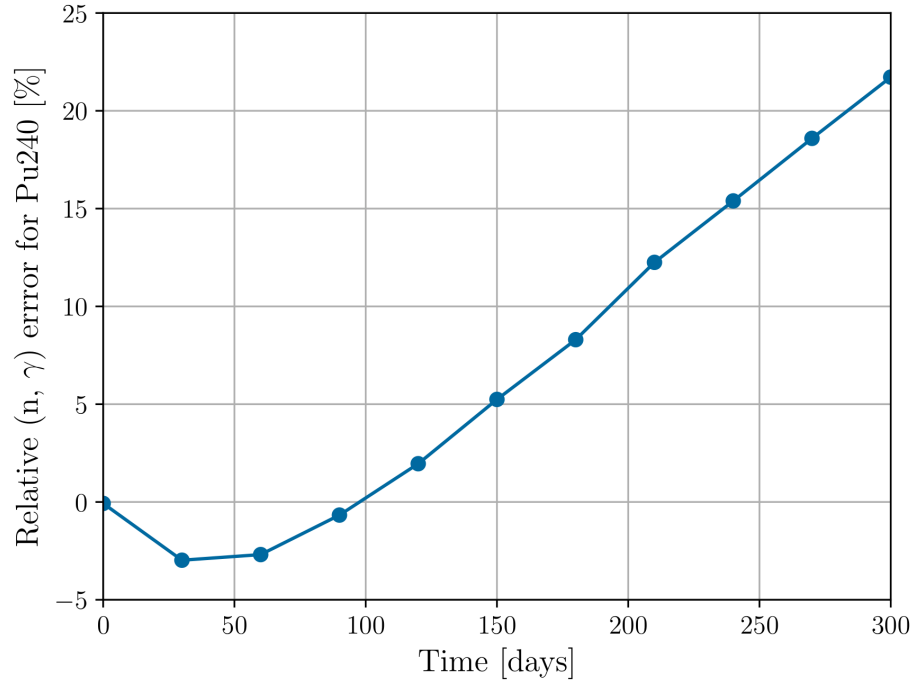
**Figure 2**. *Relative ^{240}Pu (n, γ) reaction rate error using constant cross sections and 30-day time steps.*

The error in ^{241}Pu concentration propagates to daughter nuclides that are related to the amount of ^{241}Pu, such as isotopes of Cm and Am. We observe a general trend that the less abundant nuclides have higher concentration error relative to transport-coupled depletion. The less abundant actinides have increased sensitivity to variations in production. The more abundant nuclides (U, ^{239}Np, ^{239}Pu, ^{240}Pu) tend to have low concentration errors, (5% or less), whereas the less abundant actinides (isotopes of Am, isotopes of Cm, ^{241}Pu, ^{242}Pu) tend to have high (more than 10%) concentration errors depending on the nuclide of interest.

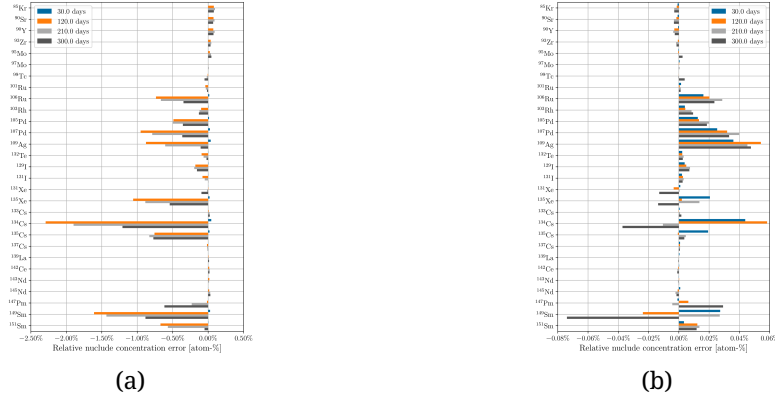(a)   (b)

Figure 3. *Relative actinide concentration error using 30-day time steps at 30, 120, 210, and 300 days of depletion for (a) constant cross sections; (b) updating cross sections.*

Figure 4a and Figure 4b show the relative error in predicted fission product concentration using constant cross sections and updating cross sections, respectively, using 3-day time steps. Figure 5a and Figure 5b show the same respective quantities for 30-day time steps. Similar to the actinides, updating the cross sections at each depletion step results in low predicted nuclide concentration errors. There is a lower concentration error across the board for many of these fission products compared to the actinides. This is due to the low concentration error of ^{235}U, which propagates to the fission products. Other actinides with higher concentration errors are orders of magnitude less abundant in the fuel than ^{235}U, so their effect on fission product concentration error is proportionally small.

Concentration errors for low-abundance nuclides using 30-day time steps follow an interesting trend where there is an initial large negative concentration error that becomes more positive over time. This behavior is due to the static cross sections and fluxes producing the same amount of these nuclides every timestep, whereas in the transport-coupled case, net production of the low-abundance nuclides decreases over time.

(a)   (b)
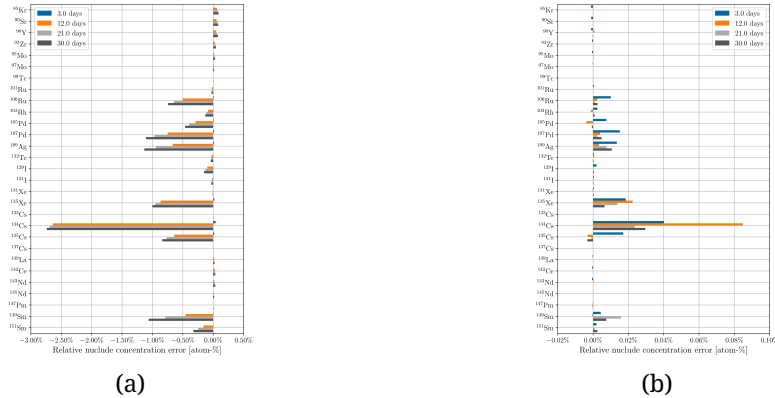
Figure 4. *Relative fission product concentration error using 3-day time steps at 3, 12, 21, and 30 days of depletion for (a) constant cross sections; (b) updating cross sections.*

(a)   (b)

Figure 5. *Relative fission product concentration error using 30-day time steps at 30, 120, 210, and 300 days of depletion for (a) constant cross sections; (b) updating cross sections.*

Repeating this analysis for both the CASMO-8 and CASMO-40 multi-group structures did not yield noticeable decreases in nuclide concentration errors for transport-independent depletion over the one-group case. Figure 6a and Figure 6b show the relative error in predicted actinide concentration using the CASMO-8 and CASMO-40 group structures, respectively, using 3-day time steps. Figure 7a and Figure 7b show the same respective quantities for 30-day time steps. It is possible in more complex models, like full reactor depletion, that the multi-group structure could become more important.

(a)   (b)

Figure 6. *Relative actinide concentration error using constant cross sections and 3-day time steps at 3, 12, 21, and 30 days of depletion for the (a) CASMO-8 group structure; (b) CASMO-40 group structure.*

(a)   (b)

Figure 7. *Relative actinide concentration error using constant cross sections and 3-day time steps at 30, 120, 210, and 300 days of depletion for the (a) CASMO-8 group structure; (b) CASMO-40 group structure.*

The transport-coupled simulations each took several hours to complete, whereas the transport-independent simulations using `IndependentOperator` took seconds to minutes to complete.

4. CONCLUSION

In this paper, we introduced transport-independent depletion in OpenMC. This method utilized pre-computed multi-group cross sections and fluxes to calculate reaction rates for a depletion calculation. The new method is much faster than running a transport-coupled depletion simulation, albeit with a penalty to accuracy. Better accuracy will be obtained for models where the neutron flux spectrum will be constant, i.e. in fusion systems and low

power fission reactors, so this new method should be used judiciously on problems where they are applicable.

This study focused exclusively on nuclide compositions. Future work may include building machinery to monitor criticality as the fuel depletes. A good starting point may be in implementing a method similar to the method to estimate k_∞ used in [10], wherein

$$k_\infty = \frac{\sum_i \sum_j N_i \nu_j \sigma_i^j}{\sum_i \sum_j N_i \sigma_i^j} \tag{8}$$

where j indicates the type of reaction, and ν_j is the neutron multiplication of reaction j, defined as

$$\nu_j = \tag{9}$$

In OpenMC, (n, xn) reactions are not used to calculate k_{eff} (and by extension k_∞). In this case, $\nu_x = 0$ in the numerator for all x, and we would use an 'effective' absorption cross section $\overline{\sigma}_a = \sigma_a - \sigma_{(n,2n)} - 2\sigma_{(n,3n)} - 3\sigma_{(n,4n)}$.

Future work could focus on characterizing the effect of sub-regions on the concentration errors. The geometry used in this study is a simple PWR pincell. It is possible that the error due to static fluxes could be reduced by dividing the pincell into sub-regions to better capture radial and axial variations as is done in high-fidelity depletion of large reactor models. Another area for future work could be on studying the importance of energy group discretization for more complex reactor models. We did not find a difference in concentration errors between one-group, CASMO-8 group, and CASMO-40 group transport-independent depletion for the pincell model. It is possible that energy group discretization becomes more important for complex models with distinct depletion zones.

References

[1] P. K. Romano *et al.*, "A Code-Agnostic Driver Application for Coupled Neutronics and Thermal-Hydraulic Simulations," *Nucl. Sci. Eng.*, vol. 195, pp. 391–411, 2021, doi: 10.1080/00295639.2020.1830620.

[2] E. Merzari *et al.*, "Exascale Multiphysics Nuclear Reactor Simulations for Advanced Designs," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, Denver, CO, USA, 2023. doi: 10.1145/3581784.3627038.

[3] P. K. Romano, "Depletion capabilities in the OpenMC Monte Carlo particle transport code," *Annals of Nuclear Energy*, p. 15, 2021, doi: https://doi.org/10.1016/j.anucene.2020.107989.

[4] P. K. Romano, N. E. Horelik, B. R. Herman, A. G. Nelson, B. Forget, and K. Smith, "OpenMC: A state-of-the-art Monte Carlo code for research and development," *Annals of Nuclear Energy*, vol. 82, pp. 90–97, 2015, doi: 10.1016/j.anucene.2014.07.048.

[5] K. D. Huff *et al.*, "Fundamental concepts in the Cyclus nuclear fuel cycle simulation framework," *Advances in Engineering Software*, vol. 94, pp. 46–59, 2016, doi: 10.1016/j.advengsoft.2016.01.014.

[6] A. M. Bachmann, O. Yardas, and M. Munk, "An Open-Source Coupling for Depletion During Fuel Cycle Modeling," *Nuclear Science and Engineering*, vol. 0, no. 0, pp. 1–14, 2024, doi: 10.1080/00295639.2024.2393940.

[7] J.-C. Sublet, J. W. Eastwood, J. G. Morgan, M. R. Gilbert, M. Fleming, and W. Arter, "FISPACT-II: An Advanced Simulation System for Activation, Transmutation and Material Modelling," *Nucl. Data Sheets*, vol. 139, pp. 77–137, 2017, doi: 10.1016/j.nds.2017.01.002.

[8] T. Eade, B. Colling, J. Naish, L. W. Packer, and A. Valentine, "Shutdown dose rate benchmarking using modern particle transport codes," *Nucl. Fusion*, vol. 60, no. 5, p. 56024, 2020, doi: 10.1088/1741-4326/ab8181.

[9] E. E. Peterson, P. K. Romano, P. C. Shriwise, and P. A. Myers, "Development and validation of fully open-source R2S shutdown dose rate capabilities in OpenMC," *Nucl. Fusion*, vol. 64, p. 56011, 2024, doi: 10.1088/1741-4326/ad32dd.

[10] M. Lovecký, L. Piterka, J. Prehradný, and R. Skoda, "UWB1 – Fast nuclear fuel depletion code," *Annals of Nuclear Energy*, vol. 71, pp. 333–339, 2014, doi: https://doi.org/10.1016/j.anucene.2014.04.021.